A novel semi-fragile forensic watermarking scheme for remote sensing images

JORDI SERRA-RUIZ* and DAVID MEGÍAS *
Estudis d'Informàtica, Multimèdia i Telecomunicacions
Universitat Oberta de Catalunya
Rambla del Poblenou 156, 08018 Barcelona, Spain

Abstract

A semi-fragile watermarking scheme for multiple band images is presented. We propose to embed a mark into remote sensing images applying a tree structured vector quantization approach to the pixel signatures, instead of processing each band separately. The signature of the multispectral or hyperspectral image is used to embed the mark in it order to detect any significant modification of the original image. The image is segmented into three-dimensional blocks and a tree structured vector quantizer is built for each block. These trees are manipulated using an iterative algorithm until the resulting block satisfies a required criterion which establishes the embedded mark. The method is shown to be able to preserve the mark under lossy compression (above a given threshold) but, at the same time, it detects possibly forged blocks and their position in the whole image.

Keywords: Hyperspectral images, Semi-fragile watermarking, Forensics, Image authentication, Tampering detection.

1 Introduction

Remote Sensing images have gained increased attention by the research community in recent years, since new uses and applications of this area are often reported. Looking for water in remote planets (NASA missions), water pollution control, high precision farming or natural resources management, among others, are well-known uses of these images (Atkinson and Tate, 1999; Wong and Lao, 2003). The acquisition of remote sensing images involves expensive equipment like aircraft or satellites. Therefore their economic value must be preserved when a third party pays for them.

New techniques for representation, storage and distribution of digital multimedia information have been developed in the recent years. Formats like MP3 or JPEG2000, and P2P networks for file sharing, increase the distribution of all kind of files. This situation has raised the problem of managing authorized and unauthorized copying, illegal distribution through the Internet and manipulation of digital information. To prevent illegal copies and the alteration of digital files (audio or image), some methods and techniques have been developed to embed a watermark into the digital files. This watermark must be imperceptible and can used to determine the integrity of the digital files. In this authentication process, two different approaches, namely semi-fragile watermarking and robust watermarking, can be used.

Both fragile and semi-fragile watermarking schemes can be used for tampering detection and localization. In fragile schemes, all modifications are detected as tampering. Therefore any kind of lossy compression or filters can not be applied to the marked image without removing the embedded mark. On the other hand, semi-fragile schemes allow some degree of compression and small modifications of the marked images. This allows, for example, to create a compressed version of the image which can be distributed electronically (possibly with a reduced price)

** Corresponding author. Email: jserrai@uoc.edu

but maintaining the original watermark. Hence, semi-fragile watermarking makes unnecessary the marking of different versions of the same image independently and, thus, reduces the cost required to distribute different versions of the same image with different degrees of quality. If the client has access to a compressed version of the image, he or she may check the integrity of the image (discarding tampering). After that, he or she may be interested in getting access to the original uncompressed image at a higher price. For example, the schemes (Yeung and Mintzer, 1997; Fridrich, 2002) embed a watermark into an image in such a way that the embedded information is destroyed or modified if the image is tampered. It is modified or destroyed when the marked image is manipulated.

Robust watermarking methods are those for which the mark is detected after strong alterations. These methods are not used for tampering detection or localization. Robust watermarking allows different types of manipulations, including compression, filtering or geometrical distortions. Usually, robustness requires reducing the transparency of the embedded mark, *i.e.* the marked file is significantly distorted. In this case, a trade-off between robustness and perceptual quality must be achieved. Robust watermarking schemes have proven successful in order to protect images in several ways, such as resolving authoring disputes or detecting changes in the images aimed to produce a forged copy, as shown in (Lin et al., 2000; Minguillón et al., 2003).

In remote sensing imaging applications, the most useful schemes aimed to detect changes in the image are semi-fragile watermarking systems. Semi-fragile schemes are able to overcome some minor modifications, as those produced by near-lossless compression, but reveal the existence of manipulations (also referred to as "attacks" in the watermarking literature), such as copy-and-replace, or excessive information removal by means of cropping or lossy compression. Most existing semi-fragile watermarking applications can be applied on monochromatic images, and thus, can be easily extended to multispectral or hyperspectral images (remote sensing images) by processing the multiple bands of these images independently.

It is worth pointing out the difference between watermarking and hashing schemes. The watermarking process alters the original data file by modifying the content in order to embed the mark. Most schemes require a common (secret) key both at the embedder at the detector which is usually introduced to endorse the system with security features. The detection process only needs the (secret) key to determine whether the image has been altered and where. With a blind detection method, no further information is needed to detect the mark (*i.e.* the original unmarked content is not required). On the other hand, hashing methods generate a given number (*e.g.* a checksum) which is needed in the detection process. If tamper localization is required, the hashing values of different areas of the image must be generated and checked at the receiver side. Hence, the detector needs all the hashing values related to the original image. In addition, if multiple versions of the image are created (*e.g.* with different compression ratios) multiple hashing values would be required (a set of values for each different version). Consequently, the semi-fragile watermarking approach makes it possible to protect the content of different versions of a hyperspectral image using a single procedure. Furthermore, the watermarking scheme can be used even for different hyperspectral images with the same embedding key. This is not possible with hashing techniques, which require the computation of a different set of hashing values for each different version of each image.

There are previous works dealing with satellite image watermarking (Qin et al., 2004; Ho et al., 2005; Wang et al., 2005; Caldelli et al., 2006; Sal and Graña, 2008). Ho et al. (2005) use a satellite image and decompose it into two mutually orthogonal sub-fields, but only uses one band of the satellite image and only one field for watermarking purposes. Qin et al. (2004) present a semi-fragile watermarking scheme based on wavelet transforms. The edge and texture of the remote sensing image are extracted and the watermark is embedded only in the edge character. In this case, once again, only one band is marked. Caldelli et al. (2006) present a scheme to embed an authentication mark using the method described by Fridrich (2002) to mark and compress the image at the same time. This scheme uses only one band to embed the mark into each block to detect manipulations. Wang et al. (2005) present a watermarking scheme to preserve a digital content, but only uses one band of the hyperspectral image of the Indian Remote Sensing System. Finally, Sal and Graña (2008) describe an evolutionary algorithm which marks an image based on the manipulation of the discrete cosine transform (DCT) computed for each band

of the image. Tamhankar et al. (2003) describe a method to embed one mark into the hyperspectral image using the whole signature, but it does not allow compression of the hyperspectral image. This adaptive watermarking method based on the redundant discrete wavelet transform (RDWT) is a fragile scheme.

In this paper, a semi-fragile watermarking scheme specifically developed for remote sensing images is presented. This method works with all the bands at the same time and provides uniform protection for multispectral and hyperspectral imaging applications. The method can be tuned to embed the mark according to band relevance, depending on the content and the signatures (also known as the spectral reflectance curve) to be protected.

Usually, fragile or semi-fragile watermarking schemes of multiple band images also consider only one band or process each band separately (Ekici et al., 2004; Caldelli et al., 2006). It is possible to work with images using only one band for computing where and how to embed the watermark, but then, when multiband images are marked with the same method, the bands are usually marked separately or only one band or a subset of bands are marked. Note that, if the bands are marked separately, the changes in the signature curves can be uneven (some values can be increased and others decreased, for the same pixel). Hence, the shapes of the signatures may vary, which may lead to a misclassification of the image (for example, a different material could be identified in the image). Because of this, a method which preserves the shapes of the signatures is highly demanded, and this can be achieved by working with the signatures as a whole.

The method suggested in this paper uses the hyperspectral image as a whole applying a vector quantization approach. The image is segmented in three-dimensional blocks of a given size which determines the spatial resolution of the embedding and detection algorithm. For each block, a tree with an endmember (real values read by the sensor for each pixel region) of the remote sensing image is built and these endmembers are manipulated by removing the least significant bits in order to increase the robustness against possible near-lossless compression attacks. Finally, the block is manipulated using an iterative algorithm until the resulting block (TSVQ tree) satisfies some criterion. The image is modified according to a secret key which produces a different criterion for each block in order to avoid copy-and-replace attacks. This key determines the internal structure of the tree and also the resulting distortion.

This paper is organized as follows. In Section 2, the coding of remote sensing images by means of tree structured vector quantization is reviewed. In Section 3, the watermarking strategy and the mark embedding and mark retrieval processes are described. Section 4 presents the results obtained with the suggested scheme for the chosen experimental corpus, and analyses the basic parameters that determine the results. Finally, the most relevant conclusions of this work are drawn in Section 5.

2 Background

In this section, an overview of three basic concepts used in the method presented in this paper, namely remote sensing images, lossy compression and vector quantization, are briefly introduced.

2.1 Remote sensing images

Remote sensing images store information about a broad area of the surface of the Earth. The construction scheme of this information is shown in Figure 1.

Each pixel is represented by a signature, which is a set of values obtained for different frequencies of the light spectrum (bands). The signature of each pixel of the remote sensing image is related to the different materials which can be found in that area, such as water, forest or minerals. Figure 2 shows the different signatures for a light reflectance for clear lake water, turbid river water, vegetation, dry soil and wet soil, as presented in Smith (2006).

One of the most relevant problems to handle these images is their huge size. A typical hyperspectral image covering a small region of a few kilometres contains millions of pixels, and each pixel is represented by several bands, depending on the sensor type. As an example, the Airbone Visible / Infrared Imaging Spectrometer (AVIRIS) (NASA, Jet Propulsion Laboratory, 2004) images contain 224 bands and, usually, 16 bits are used for the values in each band. Images with lower resolution are often referred to as "multispectral". This is the case for Landsat images (NASA, U.S. Geological Survey, 1972), which use 8 bands for each pixel and 8 bits for the values

of each band. The number of bands in a remote sensing image determines their name, that is, multi, hyper or ultraspectral.

One of the techniques to reduce the large amount of data storage is to apply a lossy compression method (Aiazzi et al., 2006; Mielikainen and Toivanen, 2006). "Lossy", means that, once the input image has been encoded and later decoded, the recovered image is not exactly the same as the original input image, but some information has been removed. Although some specific applications do not allow any kind of information removal, some information losses are allowed in many other situations. However, the noise introduced by the lossy compression process must be kept below a given threshold to avoid damaging relevant information.

2.2 Lossy compression of remote sensing images

Lossy compression methods remove information which is not significant for image reconstruction. This is the key issue in lossy image compression, because the information removed should depend on the user purposes. Several criteria for image quality can be defined, as described in Cristophe et al. (2005), depending on the desired goal. Preliminary experiments (Minguillón et al., 2000a,b) show that it is possible to achieve relatively high compression ratios without removing critical information.

Figure 3 shows a simplified block diagram of the components of the encoder in a typical lossy compression system. It mainly consists of five basic stages. In fact, most systems include a pre-processing stage (P) where, if needed, a colour model conversion or a dimension reduction is performed. Then, firstly, a transform (T) is applied to the input data in order to obtain de-correlated coefficients and a higher compactness of energy in a few coefficients. Secondly, a quantization stage (Q) removes information considered unnecessary for the user purposes. Thirdly, a bit plane encoding (BPE) is applied to account for the significance of the quantized coefficients. Finally, an entropy coding scheme (EC) is used to reduce the amount of bits needed to send the significant quantized coefficients through the transmission channel. At the receiver side, the decoder performs the inverse operations in reverse order. The overall goal is to produce a recovered image as close as possible to the original image I while preserving the bit rate needed to transmit I^* as low as possible, maximizing a quality criterion.

The general coding scheme must be adapted to the particular characteristics of the source, in order to maximize both the compression ratio and the image fidelity. In this case, the most important issue is that the remote sensing images are 3D, where two dimensions are spatial but the third one is spectral. An ideal compression method would take advantage of this fact, trying to exploit both spatial and spectral redundancy. Coding each band separately is suboptimal as spectral redundancy is not exploited, while applying 3D coding schemes does not take into account the difference between spatial and spectral dimensions. Several authors have proposed 3D transformations to decorrelate spatial and spectral redundancy. For example, Motta et al. (2005) present a survey on hyperspectral image compression. In this paper, vector quantization is used for lossy compression, and all bands are processed at the same time.

2.3 Vector Quantization and Tree Structure Vector Quantization

Vector Quantization (VQ), as described in Gersho and Gray (1992), makes it possible to compress an image in an optimal manner from the Shannon's rate-distortion theory point of view. As detailed in (Gersho and Gray, 1992; Gray and Neuhoff, 1998), Shannon (1948) showed that, given a coding rate, the least distortion achievable by vector quantisers of any kind is equal to a function, subsequently called the Shannon distortion-rate function, which is determined by the statistics of the source and the measure of distortion. Shannon's rate-distortion theory establishes the minimal amount of information which must be communicated over a channel so that the source can be approximately reconstructed at the receiver without exceeding a given distortion. Thus, applying VQ compression systems, the resulting image minimizes (locally) distortion for a given compression ratio. However, VQ compression can be computationally prohibitive. Hence alternative methods known to be suboptimal need to be explored for practical applications.

Tree Structured Vector Quantizer (TSVQ) is a suboptimal strategy which works starting with an initial centroid

as the codebook, that is, a tree with a single leaf, and then a quality criterion is applied (Mean Square Error). If there is room for improvement, the leaf with higher distortion is split into two similar centroids, and then the Linde-Buzo-Gray algorithm, described in (Linde et al., 1980), is applied for computing the new centroids. This process is repeated until a general quality criterion is achieved or when all leaves contain only equal samples within the same leaf, meaning that a perfect tree T has been built . For large images, when the number of training vectors is also large, the resulting tree is usually quite deep, although it might be very unbalanced. Nevertheless, the number of possible subtrees is large enough to explore the possibilities of finding feasible subtrees for watermark embedding.

Finally, the original image is coded using the selected subtree, replacing each original vector by the closest centroid, that is, by the centroid representing all the elements in the leaf where the original vector falls. This selection is performed starting from the root of the tree, and choosing the closest centroid until a leaf is reached. Unlike the generation of the initial tree T, coding the tree *selected* is a very fast operation which can be performed very efficiently.

3 Multiple band watermarking scheme

The watermarking scheme presented in this paper is described in the following sections. In Subsection 3.1, the lossy compression and watermarking are described, while Subsection 3.2 shows how to detect tampered marked images.

3.1 Mark embedding process

Let's consider an original three dimensional hyperspectral image I of size $M \times N \times b$, where b stands for the number of bands. This paper uses AVIRIS images, but the same method can be applied to any kind of multi, hyper or ultraspectral images.

Pre-processing step

An optional pre-processing step of the watermarking scheme suggested here is to compress and decompress the original image with JPEG2000 (Taubman and Marcellin, 2002), *e.g.*using the KaKaDu software (Taubman, 2007) with 14 bits per pixel (bpp) as compression parameter, in such a way that the noise of remote sensing image sensors is removed and the less significant information of the signature is also removed. If this pre-processing step is applied, the robustness of the scheme is increased, but the Peak Signal-to-Noise Ratio (PSNR) of the marked image decreases. Nevertheless, this step helps reduce the noise of the sensors and any potential spurious values in the data are removed.

Band selection, LSB extract and block division

For hyperspectral and ultraspectral images, with a large number of bands and a huge number of pixels, we propose to group bands according to a reordering criterion, creating different spectral signatures which will be processed separately, or to select a set of bands with a significant information to be precessed, following the recommendations given by Gersho and Gray (1992). In this paper, several hyperspectral images have been chosen to test the scheme. The experiments have been performed for set of bands (16 bands selected out of 224) of AVIRIS images. The selected bands are equally spaced in the wavelength axis.

After band selection, the resulting image has 16 values (one for each band) of 2 bytes per pixel. To increase the robustness of the process against compression attacks, the n least significant bits (LSB) of each pixels are removed (and will be restored afterwards). The larger the number n of extracted LSB, the more robust the watermarking scheme will become. Increasing the number of LSB increases the robustness, but decreases the quality of the marked image. However, the number of extracted LSB must limited such that each value has enough information to build TSVQ trees as detailed below. Nevertheless, these LSB are kept to generate the final marked image, in such a way that the final marked image has 16×2 bytes (in this case, 16 is the number of bands) by pixel. The LSB recovery process is detailed below.

Finally, the image I is segmented in blocks $W \times H \times b'$, where $W \leq M$ and $H \leq N$ are the size in pixels of

each block, and $W \times H$ vectors are created by grouping all pixels and spectral values (bands), and $b' \leq b$ is the number of selected bands. In this paper, we have used $b = 16$. This block division makes it possible to detect specific tampered regions in attacked images.

Block mark embedding

Figure 4 summarizes the selection of the bands, the segmentation of the image in blocks, and the construction of the TSVQ vectors. For an image of 512×512 pixels, there are $262\ 144$ signatures with 16 values, one for each band, which is a reasonably large value for vector quantization purposes.

The vectors for each block B are replaced by very similar values (block) B' obtained from the leaves of a TSVQ tree, chosen from a family of trees, with minimal distortion and enforcing a particular property which will be checked at the detector side. TSVQ lossy compression is thus used to generate a similar image (block) B^*. The relevant steps of the TSVQ process applied here are preprocessing (P), transform (T) and quantization (Q), as described in Section 2.2.

Using the TSVQ algorithm described in Gersho and Gray (1992), a complete tree of centroids is built, containing all the possible subtrees for compressing the original image block. Then, the BFOS algorithm of Breiman et al. (1984) prunes the generated tree with the selected criterion to obtain all the subtrees in the Compression ratio-Distortion curve, namely the convex hull. A parameter of the resulting TSVQ tree determines the subtree that is required to obtain the specific compression of the image block, for example the entropy or compression ratio itself and selects only one subtree from the thousands of possible subtrees. The BFOS algorithm ensures the best possible subtree with the lowest compression ratio for the selected criterion, in the optimal ratio-distortion curve generated by all the optimal subtrees.

As a possible improvement of the proposed method, if the maximum difference between the original vectors and the centroid of the leaf where they fall into is known, any modified version of the centroid between the original vector and the selected centroid could be used for obtaining a modified image block.

In order to exploit both spatial and spectral (3D) redundancy, some authors propose to use a Discrete Wavelet Transform (DWT) on each band and then a vector quantization step for each new spectral signature using a multiresolution approach. We have followed a similar approach, but we use the original pixels, without any transformation (apart from the LSB extraction step), because we do not try to maximize the compression ratio, but to obtain similar images for embedding information for watermarking purposes.

The division of the original image into blocks is performed following the recommendations in Raudys (1997), where it is shown that the ratio between the number of available samples and vector dimensionality should be at least 30 in order to minimize any statistical bias caused by an insufficient number of samples. Therefore, as in our case the dimension of the signatures is 16 (number of selected bands), we need at least 16×30 different samples for compression and classification purposes. For this reason, the regions must have at least 480 samples. If we use square regions with powers of two sizes, the minimum size of these regions is 32×32 (which results in $1\ 024$ samples). We have used regions of 64×64 pixels in order to reduce the number of blocks. Then, the original 512×512 image has been divided into 8×8 small regions, and these 64 small regions are marked separately.

At this point, it is important to note that the mark of each block is determined by the criterion selected in each block. However, the compression ratio does not take into account the individual bands but the signatures as a whole, since we are using a vector quantization approach. Hence, this proposal is different from other semi-fragile watermarking methods in the literature (Rey and Dugelay, 2002; Ekici et al., 2004) which process each band separately.

As manipulations to the marked image are concerned, a large modification in any single band or small number of bands is considered unacceptable in remote sensing images, because these attacks introduce an uneven change in the spectral signature. Only modifications affecting the whole signature are accepted, such as lossy compression, but only up to a certain degree.

In Table 1, an example of the different subtrees generated during the pruning algorithm is described. The columns of this table represent the ratio between the distortion and the compression ratio (λ), the compression ra-

tio, the distortion, the maximum and minimum depth of the subtree, the number of nodes of the subtree and the entropy of the subtree. All these values are given for each possible subtree which can be generated by pruning the original one (denoted by index 0). These measures are used as the basic criterion for stopping the pruning algorithm and determining the specific subtree which will be used in the marked image block. A simple but efficient criterion is to stop when the maximum depth achieves a desired value. Even with simple criteria like this, it is possible to generate complex subtrees which are impossible to reproduce by manipulating the original image. Table 1 shows an example of the subtree process selection by a entropy criterion. In this case, the criterion is 8.64 and the subtree 805 is selected.

Each block of the original image is compressed with a different compression ratio, according to the selected criterion. This new block is processed with the TSVQ process again, using the same parameters as in the first iteration, and this process is repeated until the generated block has the desired properties.

The secret key is used in this point, since a different criterion is selected for a each block of 64×64 pixels. To detect possible tampering attacks, such as copy-and-replace, or pasting one part of another image into the marked image, a pseudo-random sequence is chosen to determine the values used as criteria to select the compression subtree in the pruning algorithm. Thus, it is much more difficult to find a pattern revealing the watermarking scheme properties, thus reducing the possibilities of manipulating an image or modifying it using another region of the same image. A key is required to watermark images, which is used to generate the pseudo-random sequence. Hence, each block of the marked image has a different mark (property), to allow forensic process over the marked image. This key is the seed of a Pseudo-Random Number Generator (PRNG) and will be required in the detection process. Nevertheless, the proposed method does not need the original image for tampering detection, thus it can be considered blind.

The entropy of the tree is used to select which tree is used to build the marked block. For each block, the pseudo-random sequence determines the entropy of the compression tree. However, not all the values of entropy are possible, but only a set of these values are chosen (the entropy is quantized).

Finally, the BFOS algorithm generates a table with all the possible subtrees which are in the convex hull, minimizing distortion for a given compression ratio. Usually, both compression ratio and MSE are used for generating the convex hull, but the BFOS algorithm may be used with any other criteria which might be more suitable for watermarking purposes or for joint compression and watermarking as detailed in Caldelli et al. (2004).

Once it is determined which subtree is used to generate the modified block, the resulting watermarked block is constructed with the centroids of the selected subtree which represents the embedded mark.

The number (n) of LSB bits that are extracted from the original image is a parameter of the scheme which determines the robustness of the marked image against compression attacks. Therefore, a simple compression of marked image is accepted, because a small ratio compression modifies only a few LSB of signature vectors. The n LSB bits of the marked pixels are restored as follows (Figure 5):

1. If the new value of the pixel is lower than the unmarked value, the LSB are filled with ones.
2. If the new value of the pixel is greater than the unmarked value, the LSB are filled with zeroes.
3. If the new value of the pixel is identical to that of the unmarked value, the LSB are restored from the original pixel.

Let LSB$'$ be the new value of the LSB bits. The restored bits are computed using equation 1:

$$\text{LSB}' = \begin{cases} 1s : B^*_{n,m,b} < B_{n,m,b} \\ 0s : B^*_{n,m,b} > B_{n,m,b} \\ \text{LSB} : B^*_{n,m,b} = B_{n,m,b} \end{cases} \quad (1)$$

Summary

The following steps (see Figure 6) summarize the mark embedding process:

0. Pre-processing: compression and decompression the original image with JPEG2000 (KaKaDu Software).
1. Band selection and block construction of the image to be marked (as shown in Figure 4).

2. Extract n LSB for each pixel component.

3. Choose the seed of Pseudo-Random Number Generator (PRNG) and initialize it.

4. Choose the mark property (*e.g.* entropy).

5. For all blocks

   5.1 Generate a number with the PRNG and check it is not repeated for a previous block.

   5.2 Choose the entropy according to the number generated in step 5.1.

   5.3 Select the tree with entropy value chosen in step 5.2 (see Table 1).

   5.4 Generate the image using the tree obtained in step 5.3.

   5.5 Check the property selected in 5.2. If not checked go to 5.3 with the image generated in 5.4.

   5.6 Restore the LSB extracted in 2 minimizing the difference of the pixel values.

6. Join all the blocks (and the bands not selected in step 1) to build the final marked image.

3.2 Forgery detection

The mark detection process for a (possibly forged) image, shown in Figure 7, is analogous to the mark embedding process, and it only requires the seed of the pseudo-random sequence. First of all, the same bands used for embedding must be extracted from the marked image. Then, it must be divided into the same blocks to determine if the mark is present or if they have been modified (above a given threshold). As detailed in the figure, the LSB must be extracted from the (assumed) marked block. For each block and the corresponding property for that block (computed from the PRNG), the detection process is performed.

The detection of a modification of the image can be performed by checking if the same criterion used in the mark embedding scheme is satisfied. Thus, the TSVQ tree is constructed for each block and the criterion is checked. If the block verifies the criterion then the area is assumed to be authenticated. Otherwise, the block is detected as forged. Therefore, with this method it is possible to inspect and locate tampered regions in a marked image.

4 Experimental Results and Comparative Analysis

This section presents the results obtained with the method described above and also compares the suggested scheme with other watermarking schemes in the literature.

4.1 Experimental Results

As described in section 3.1, a hyperspectral AVIRIS image (Figure 8) has been used to evaluate our detection system for tampered images. For evaluation purposes, we measure the influence of the embedding process on image quality and an example of the positive detection of a copy-and-replace attack is shown.

The chosen image is that of a cuprite area of Nevada. This image contains relevant information about minerals which is essential to protect against malicious tampering attacks. In a cuprite area, there are different minerals with different reflectance which will lead to different economical profits. Because of this, a potential buyer of the hyperspectral image, needs too be confident about the authenticity and integrity of the image in case he or she wants to obtain economical benefits by exploiting the corresponding area. This image has been taken from a $12\,280$ metres $\times$ $40\,960$ metres area. Although the suggested scheme can be applied to any image size, we have selected a 512×512 pixel image size, which represents a square-shaped area of $10\,240$ metres of width and height. This area is then divided into 8×8 blocks and each block contains 64×64 pixels (representing $1\,280 \times 1\,280$ square metres of the area). Each pixel is represented by a 2-byte (16-bit) integer. However, the two Most Significant Bits (MSB) are always zero. Hence, it can be considered that the pixels are actually represented by 14 bits.

The AVIRIS image of this cuprite zone has 224 different bands. A subset of 16 bands has been chosen to embed the mark. The subset of bands can be selected according to different criteria, such as random selection, equally spaced bands in the frequency axis, atmospheric criteria (reflectance is different at difference frequencies due to

atmospheric reasons) or materials of the image area (some frequencies provide better light reflectance for some materials). In this experiment, for simplicity, 16 equally spaced values in the frequency axis have been chosen. We have selected 1 out of each group of 16 bands in the original hyperspectral image. In particular, the selected bands are the following: 4, 18, 32, 46, 60, 74, 88, 102, 116, 130, 144, 151, 172, 186, 200 and 215. The 158 and 214 bands have some errors in the values read by the sensor (*e.g.* the band 158 has all values assigned to zero). Because of this, the 151 and 215 bands have been chosen instead. Figure 8 shows a greyscale bitmap corresponding to three of these selected bands.

In a first experiment, we have used $n = 2$, *i.e.* the 2 LSB of the original image have been extracted and they were replaced later on by the value which minimizes the distance from the original image to the marked one. This LSB-removal process produces robustness against a compression attack (such as a JPEG or a JPEG2000 coder).

The mark embedding process affects the image quality of the signature of the image of the selected bands. The PSNR is a measure of distortion commonly given in decibels (dB), using a logarithmic scale for the results. The PSNR of the marked hyperspectral image computed for all the 224 bands is **77.336 (dB)**, and the percentage of modified pixels is 0.96% (only $563\,502$ out of the $58\,720\,256$ pixels are modified by the embedding process). The PSNR (equation 2) is calculated with the maximum value of pixel image. For this particular image, the maximum value is $12\,500$, therefore the PSNR has been calculated with the value $2^{14} - 1 = 16\,383$ instead of $65\,535$.

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) = 10 \cdot \log_{10}\left(\frac{16\,383^2}{\text{MSE}}\right) \tag{2}$$

The PSNR for each of the modified bands is shown in Table 2, where "PMP" stands for "Percentage of Modified Pixels" and "Adi" stands for "Average difference" of the modified pixels. Notice that PSNR is quite high, specially for central bands containing more information, and it is slightly worse for the last few bands, although the obtained values are still reasonable. On average, the PSNR for the different modified bands is **66.257** dB. The number of modified pixels is very small, only 13.43% of the $262\,144$ pixels per modified band are changed with an average difference of 16.84 (where the range of pixel values is from $1\,200$ to $12\,500$).

The distribution of the error introduced in the embedding process is shown in Figure 9 only for the modified pixels (note that 99% of the pixels are not modified). As reported above, the average of the absolute value of the error (difference between the original and the marked pixel) for the modified pixels is 16.84. Although the differences are always integer numbers, note that the average of the absolute values of these differences, computed for all the modified pixels, is a rational number.

The difference between the original signature (for one pixel) and the marked signature is shown in Figure 10. The marked signature is shifted 200 units down only to show the difference between the two signatures. The figure shows that the two signatures are almost identical. Hence the classification methods work identically either with the original or the marked signature, since the modification only affects 0.96% of the pixels with a average difference of 16.84. In this figure, we have taken into account the errors due to the data acquisition. For instance, the original image has some zero or $65\,535$ values due to sensor errors. These values have been removed to show the correct curves of the original and marked signatures.

As shown in Table 3, the number of removed LSB affects the difference between the pixels values after mark embedding. This table shows, from left to right, the image used in the experiment, the number of removed LSB, the PSNR of the marked bands, the PSNR of the whole image, the average of the differences (in absolute value) of the modified pixels (the original value minus the marked value), the average deviation of these differences and the number of modified pixels. The PSNR results for the Cuprite image for $n = 2$, 3 and 4 are given in the first few rows. It can be seen that, by removing 4 LSB ($n = 4$) the PSNR increases, on average, to 67.58 dB, but the average difference raises up to 17.20 instead of 16.84 (the average difference for $n = 2$). This is because the mark embedding scheme affects the bits from the 5th LSB to the MSB if 4 LSB are removed, leading to better robustness against compression but increasing the difference between the modified and the original pixels.

Note that the PSNR increases when the n (the number of removed LSB) increases. The reason for this behaviour is that the TSVQ procedure modifies less pixels when n

increases: with $n = 2$ the scheme modifies 0.96% of all pixels, but with $n = 4$ it modifies only 0.71%. Increasing n reduces the number of significant bits used for building the vectors and reverts on smaller TSVQ trees and, thus, the TSVQ procedure modifies a smaller amount of pixels leading to an increased PSNR (which is counter-intuitive). However, the larger n is, the larger the difference between the original pixel value and the marked value becomes. The difference is small on average, but in a particular band it can increase to 25.32. The number of significant bits $(16 - n)$ can not be reduced indefinitely, since the number of vector values would not be enough to build the classification trees.

As discussed in Section 1, semi-fragile watermarking schemes make it possible to produce compressed versions of the image preserving the embedded information. These versions can be distributed electronically at different prices according to the degree of quality, and the buyer can still determine the authenticity of the marked image. For this reason, the robustness of the scheme has been measured against compression attacks. Compression attacks are defined as a JPEG2000 compression and decompression process, for 3:1 and 2:1 compression ratios. These attacks are supported and do not affect the mark embedded in the image (for $n = 2$). Higher compression ratios can remove the mark but also remove some relevant information of the images. This establishes a trade-off between quality and robustness (or fragility). If larger compression ratios are required, n can be increased (to 3 or 4, for example).

The PSNR results given above have been obtained for the Cuprite image with 14 bpp, and hence the PSNR is obtained by dividing $\left(2^{14} - 1\right)^2 = 16\,383^2$ by the MSE. Other methods use images with 8 bpp and thus, their PSNR results are obtaining by dividing $\left(2^8 - 1\right)^2 = 255^2$ by the MSE. In principle, the PSNR could be greater with 14 bpp than with 8 bpp. In order to overcome this drawback, a set of experiments have been carried out with different images:

- The Cuprite image reduced to 8 bpp (the 6 LSB have been suppressed).
- The Indian Pines AVIRIS image, which is commonly used in other papers, like Sal and Graña (2008), with 14 bpp (two bytes per pixel, but the two most significant bits are always 0).
- The Indian Pines image reduced to 8 bpp.

For the 8 bpp images, the values of n are 0, 1 and 2. Larger values are not advisable since that would mean removing 3 or more LSB out of only 8.

The PSNR results obtained for these images are also shown in Table 3. As expected, the PSNR for 8 bpp images is a bit lower than that obtained with 14 bpp images. With 8 bpp, the PSNR is around 60 dB for the marked bands and 70 dB for the whole image. For 14 bpp, the results are around 70 dB for the marked bands and 80 dB for the whole image. It can be noticed that the results are remarkably similar between the two chosen images when the number of bpp is the same.

Table 4 shows some compression attacks and the robustness of the scheme for different values of n (for the Cuprite image with 14 bpp). This table provides the average and the maximum difference for the pixel values between the attacked file and the marked one. Note that the scheme allows some compression ratio. In general, the parameter n determines the robustness of the scheme, since pixel deviations up to $2^n - 1$ will be allowed. With $n = 2$, the compression is only permitted to 8 bpp (the original image has 16 bpp) since the maximum difference in the pixel values is $3 = 2^2 - 1$. With $n = 3$ and $n = 4$, the compression is allowed up to 7 and 6 bpp respectively, since they allow deviations of up to 7 and 15 in the pixel values (also respectively). This example illustrates how the parameter n can be used to tune the robustness/fragility trade-off of the suggested method.

Finally, a random block selection and replacement into the image has been performed into the marked image. All the copy-and-replace attacks have been detected by the scheme. Figure 11 shows the original image and the marked image with one specific key and $n = 2$. Notice that no artifacts are visually detectable. On the other hand, Figure 12 shows a tampered image and the tamper location as detected by the proposed scheme. In this image, it is possible to see a forest region which has been copied and replaced into a mineral region, next to the forest. This forgery has been perfectly detected by the suggested scheme. In addition, no false positive forgeries have been detected in the performed experiments.

4.2 Comparative analysis with other watermarking schemes

This Section presents a comparison between the suggested scheme and other watermarking systems for remote sensing images. Some previous works have been selected and compared with the proposed method. All of them are blind–except Tamhankar et al. (2003)–, semi-fragile and allow some level of compression. The selected method can be used for tampering detection, but not all of them report the tamper localization.

Table 5 shows the results obtained with each method. In the first column, the chosen method is referenced. The second column shows the type of the remote sensing image used in the corresponding method. The third column describes whether the method is applied to a single band, a subset of bands or the whole signature. The fourth column reports the PSNR obtained with each method (if available). Finally, the fifth column indicates whether the method can be used for tamper localization, reporting either the size of the identified tampered area or "No" if it can applied only for tamper detection (not localization).

It can be noticed that the proposed scheme provides tamper localization, works with the whole signature and yields extremely high image quality, since the PSNR overcomes that of the other schemes: around 70 dB for 8 bpp images against 55 dB for the best of the other schemes. Even if we consider the PSNR only of the 16 marked bands, the proposed scheme yields PSNR around 60 dB, still better than that of the other schemes. The different methods have not been tested for exactly the same images for several reasons: some of them do not provide exact values for the tuning parameters, some work for different types of images (RGB against hyperspectral), some use sophisticated techniques like genetic algorithms which require expertise in order to implement them, etc. However, the results given in Table 3 show that the proposed scheme yields similar PSNR for different images (as far as the number of bpp is the same). Thus, the comparison for 8 bpp images provided in Table 5 is fair enough.

Among the methods which are applied to hyperspectral images, note that Sal and Graña (2008) is applied to each band separately, which means that the signature is not modified evenly and the curve can be significantly altered. The scheme described by Sal and Graña (2008) can be used to detect tampering, but not the localization of manipulated signatures. Finally, although the scheme of Tamhankar et al. (2003) works with the whole signature of a set of pixels, it does not provide information about tamper localization either. As the size of the tampering localization is concerned, (Qin et al., 2004; Ho et al., 2005; Caldelli et al., 2006) make it possible to identify smaller tampered areas compared to the proposed scheme, but those methods must be applied to each separate band and yield worse PSNR results.

5 Conclusions

In this work, a semi-fragile watermarking method for multi and hyperspectral data based on tree structured vector quantization and compression is presented. The method uses the information in all the bands at the same time, and thus, it takes advantage of both spatial and spectral redundancy for marking purposes. Basically, the original image is segmented in three dimensional blocks and a tree structured vector quantizer is built for each block together with a LSB extracting process. The original block is replaced by a new one generated by substituting each original vector by the closest centroid in the selected subtree. This process is repeated until a certain stopping criterion is satisfied. Each block generates a different subtree and a secret key is used to avoid copy-and-replace attacks between blocks.

The results show that copy-and-replace attack of a region of the image is detected by the watermarking scheme, whereas near-lossless (JPEG2000 or JPEG) compression can be applied up to ratios 3:1 preserving the mark in the compressed image. The detection process is a simple one, since a tree is built for each block and the selected tree property is tested. If a block satisfies such a property, then it has not been forged, otherwise the detection process reports tampering.

An operational setting of the method described in this paper would be as follows. An image distributor wants the hyperspectral image of a given area and contacts an image provider (*e.g.* a space agency) to buy it. The image provider would obtain the hyperspectral image, apply the embedding process described in Section 3.1 (with specific keys chosen for the image and/or the distributor) and provide the distributor with the requested image and the key. In addition, the image provider will also offer the

mark detection software for any interested party. The distributor can offer (re-sell) different compressed versions of the image to different buyers. Apart from the marked image and the secret key, the buyers would request the image provider for the mark detection software to check the authenticity of the image. The buyers would feed the software with the purchased image and the secret key. The mark detection software would be the same for all images, distributors and buyers (with the only possible change of the secret key) and it would report if the image has been forged and, if it is the case, the tamper locations. Hence, once validated, the buyers can be confident about of the authenticity of the image.

Future research lines in this subject include the study of optimal subtree selection criteria for pruning purposes, in order to reduce the size of the blocks and, at the same time, to detect the most likely modified pixels instead of complete blocks, increasing the resolution of tampering detection. A further evaluation of the proposed scheme for assessing robustness-fragility, capacity and the impact on classification accuracy is under study.

Acknowledgements and disclaimer

This work is partially supported by the Spanish Ministry of Science and Innovation and the FEDER funds under the grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER-INGENIO 2010 CSD2007-00004 ARES.

References

Aiazzi, B., Alparone, L., Baronti, S., Lastri, C., and Santurri, L. (2006). *Hyperspectral Data Compression*, chapter Near-Lossless Compression of Hyperspectral Imagery Through Crisp/Fuzzi Adaptive DPCM, pages 147–178. Springer Science+Business Media, Inc.

Atkinson, P. M. and Tate, N. J. (1999). *Advances in remote sensing and GIS analysis*. Wiley.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group.

Caldelli, R., Filippini, F., and Barni, M. (2006). Joint near-lossless compression and watermarking of still images for authentication and tamper localization. *Signal Processing: Image Communication*, 21:10(10):890–903.

Caldelli, R., Macaluso, G., Barni, M., and Magli, E. (2004). Joint near-lossless watermarking and compression for the authentication of remote sensing images. In *Proceedings of the 24th International Geoscience and Remote Sensing Symposium*, volume 1.

Cristophe, E., Léger, D., and Mailhes, C. (2005). Quality criteria benchmark for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2103–2114.

Ekici, O., Sankur, B., Naci, U., Coskun, B., and Akcay, M. (2004). Comparative assessment of semifragile watermarking methods. *Journal of Electronic Imaging*, 13(1):209–216.

Fridrich, J. (2002). Security of fragile authentication watermarks with localization. In SPIE, editor, *Security and Watermarking of Multimedia Contents*, volume 4675, pages 691–700, San Jose, CA. SPIE.

Gersho, A. and Gray, R. M. (1992). *Vector Quantization and Signal Compression*. Communications and Information Theory. Kluwer Academic Publishers, Norwell, MA, USA.

Gray, R. M. and Neuhoff, D. L. (1998). Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383.

Ho, A. T. S., Zhu, X., and Woon, W. (2005). A semi-fragile pinned sine transform watermarking system for content authentication of satellite images. *IEEE International Geoscience and Remote Sensing Symposium*, 1-8.

Lin, E., Podilchuk, C., and Delp, E. (2000). Detection of image alterations using semi-fragile watermarks. *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents II*, 3971.

Linde, Y., Buzo, A., and Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28(1):84–95.

Mielikainen, J. and Toivanen, P. (2006). *Hyperspectral Data Compression*, chapter Lossless Hyperspectral Image Compression via Linear Prediction, pages 57–74. Springer Science+Business Media, Inc.

Minguillón, J., Herrera-Joancomartí, J., Megías, D., and Serra-Sagristà, J. (2003). Evaluation of copyright protection schemes for hyperspectral imaging. In *Proceedings of Image and signal processing for remote sensing IX*, volume 5238, pages 512–523, Barcelona, Spain.

Minguillón, J., Pujol, J., Serra, J., and Ortuño, I. (2000a). Influence of lossy compression on hyperspectral image classification. In *Proceedings of Data Mining'2000*, pages 545–554, Cambridge, UK.

Minguillón, J., Pujol, J., Serra, J., Ortuño, I., and Guitart, P. (2000b). Adaptive lossy compression and classification of hyperspectral images. In *Proceedings of Image and Signal Processing for Remote Sensing VI*, volume 4170, pages 214–225, Barcelona, Spain.

Motta, G., Rizzo, F., and Storer, J. A. (2005). *Hyperspectral Data Compression*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

NASA, Jet Propulsion Laboratory (2004). Airborne visible / infrared imaging spectrometer (aviris). `http://aviris.jpl.nasa.gov`.

NASA, U.S. Geological Survey (1972). Landsat program. `http://landsat.gsfc.nasa.gov`.

Qin, Q., Wang, W., and Chen, S. (2004). Research of digital semi-fragile watermarking of remote sensing image based on wavelet analysis. *IEEE International Geoscience and Remote Sensing Symposium*, 1-7.

Raudys, S. J. (1997). On dimensionality, sample size, and classification error of nonparametric linear classification algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):667–671.

Rey, C. and Dugelay, J.-L. (2002). A survey of watermarking algorithms for image authentication. *Journal on Applied Signal Processing*, 6:613–621.

Sal, D. and Graña, M. (2008). *Studies in Computational Intelligence*, volume 133/2008, chapter A Multiobjective Evolutionary Algorithm for Hyperspectral Image Watermarking, pages 63–78. Springer Berlin / Heidelberg.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423 and 623–656.

Smith, R. B. (2006). Introduction to hyperspectral imaging. Available at `http://www.microimages.com/getstart/hyprspec.htm`. (Accessed October 10, 2012).

Tamhankar, H., Bruce, L., and Younan, N. (2003). Watermarking of hyperspectral data. *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International*, 6:3574–3576 vol.6.

Taubman, D. (2007). Kakadu JPEG-2000 encoder v5.2. `http://www.kakadusoftware.com`. (Accessed October 10, 2012).

Taubman, D. and Marcellin, M. (2002). *JPEG2000: Image compression fundamentals, standards and practice*. Kluwer Academic Publishers (KAP).

Wang, X., Guan, Z., and Wu, C. (2005). *Advanced Data Mining and Applications*, volume 3584/2005 of *LNCS*, chapter A Novel Information Hiding Technique for Remote Sensing Image, pages 423–430.

Wong, F. C. and Lao, N. Y. (2003). Economic value of remote sensing imagery for agricultural applications. In *Proceedings of the Aerospace Conference*, volume 8, pages 3815–3829.

Yeung, M. and Mintzer, F. (1997). An invisible watermarking technique for image verification. In *International Conference on Image Processing, IEEE*, volume 2, pages 680–683, Santa Barbara, CA.

Table 1: Example of the values obtained with the BFOS algorithm.

Subtree	λ	Compression rate	Distortion	Num. nodes	Min depth	Max depth	**Entropy**
0	0.222	13.303	0.0000	6 813	6	24	11.651
1	0.666	13.302	0.0002	6 811	6	24	11.649
2	0.666	13.301	0.0006	6 809	6	24	11.649
3	0.833	13.301	0.0011	6 807	6	24	11.648
4	0.889	13.300	0.0019	6 805	6	24	11.647
5	1.000	13.299	0.0026	6 803	6	24	11.647
6	1.000	13.298	0.0035	6 801	6	24	11.645
7	1.111	13.296	0.0055	6 791	6	24	11.644
8	1.250	13.295	0.0063	6 789	6	24	11.643
...	...	...	...	...	...	...	...
805	10.000	9.5067	19.6232	2 867	6	17	**8.640**
...	...	...	...	...	...	...	...

Table 2: PSNR of the marked image for each band.

Band	PSNR	PMP	Adi	Band	PSNR	PMP	Adi
1	64.309	13.368	20.458	9	62.369	13.969	25.416
2	67.034	13.176	14.515	10	68.115	13.258	13.609
3	67.411	13.304	14.357	11	67.917	13.319	13.911
4	67.468	13.338	14.402	12	64.765	13.725	19.781
5	68.072	13.225	13.353	13	65.730	13.641	17.656
6	68.531	12.124	12.735	14	65.294	13.575	18.031
7	68.359	13.166	13.039	15	65.201	13.569	18.773
8	67.448	13.329	14.452	16	63.087	13.871	23.596

Table 3: PSNR values for different images and different values of n.

Image	n	PSNR (dB) Marked bands	PSNR (dB) Whole image	Adi	Diff. dev.	NMP
Cuprite (14 bpp)	2	66.25	77.34	16.84	16.83	0.96%
	3	66.50	77.57	17.07	15.70	0.87%
	4	67.58	78.42	17.20	15.72	0.71%
Cuprite (8 bpp)	0	58.74	69.93	1.14	0.39	0.46%
	1	58.22	69.17	1.26	0.52	0.42%
	2	61.06	71.56	1.43	0.76	0.17%
Indian Pines (14 bpp)	2	70.87	79.43	11.13	12.14	1.12%
	3	71.88	80.03	11.65	12.55	0.91%
	4	73.22	80.92	13.01	13.17	0.63%
Indian Pines (8 bpp)	0	60.38	70.22	1.12	0.36	0.45%
	1	60.97	69.70	1.30	0.52	0.35%
	2	65.25	72.38	1.53	0.84	0.12%

Table 4: Robustness of the scheme against compression attacks

n bits	Compression (bpp)	Mark survival	Average difference	Max. difference	NMP
2	8	yes	1.14	3	48.91%
	7	no	1.65	7	70.49%
	6	no	2.41	13	83.84%
3	8	yes	1.15	3	49.29%
	7	yes	1.47	7	70.59%
	6	no	2.43	12	84.01%
	5	no	4.13	19	91.73%
4	8	yes	1.12	3	50.23%
	7	yes	1.57	7	72.07%
	6	yes	2.42	14	81.13%
	5	no	4.19	26	91.33%

Table 5: Comparison of the proposed scheme with other watermarking systems.

Scheme	Image type	Embedding strategy	PSNR	Tamper localization
Caldelli et al. (2006)	Greyscale (1 band)	1 band	≤ 45 dB (8 bpp)	16×8 blocks
Ho et al. (2005)	Greyscale (1 band)	1 band	~ 40 dB (8 bpp)	8×8 blocks
Qin et al. (2004)	RGB	RGB	Not reported	$\sim$ Tampered area
Wang et al. (2005)	Panchromatic (1 band)	1 band	~ 55 dB (8 bpp)	No
Sal and Graña (2008)	Hyperspectral	Band by band	Not reported	No
Tamhankar et al. (2003)	Hyperspectral	Selected signatures	Not reported	No
Proposed	Hyperspectral	16 bands	~ 80 dB (14 bpp) ~ 70 dB (8 bpp)	64×64 (or 32×32)

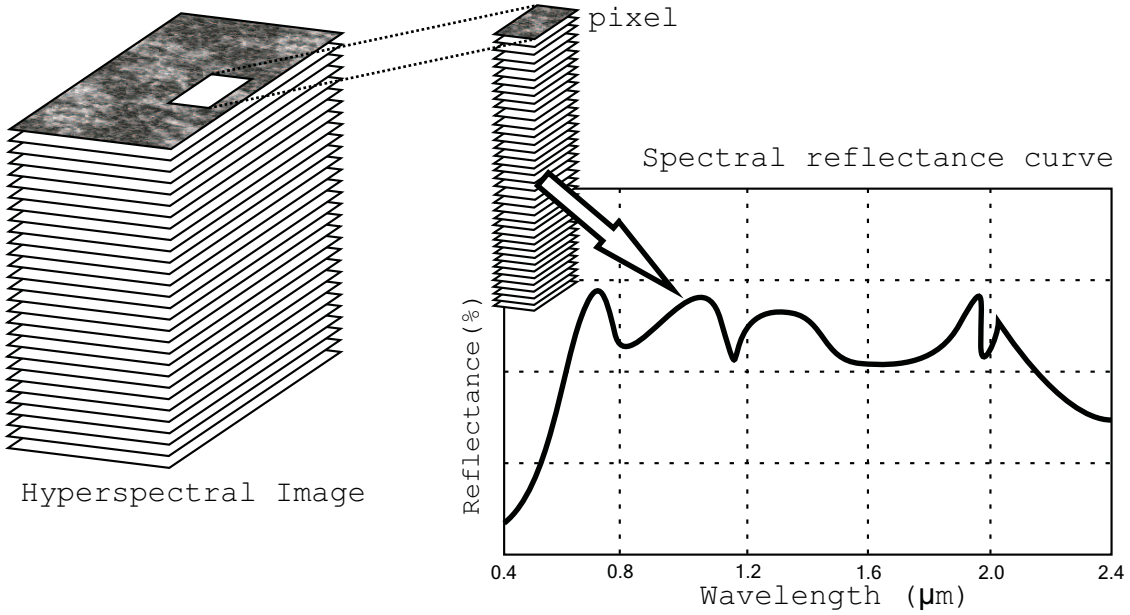

Figure 1: Example of the signature curve for a pixel.

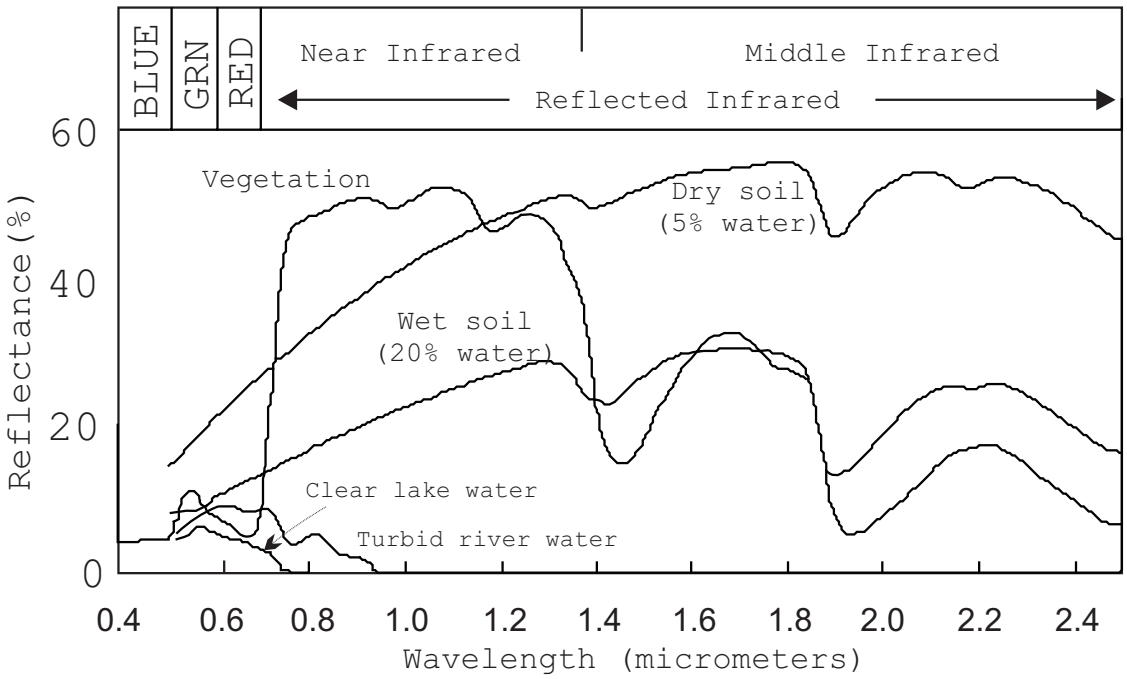

Figure 2: Sample signatures for different materials.

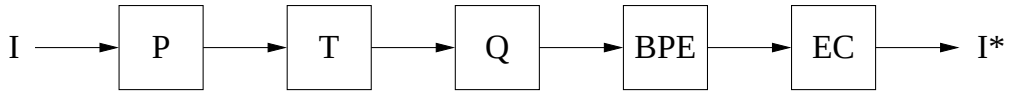

Figure 3: Encoder block diagram of a lossy image compression scheme.

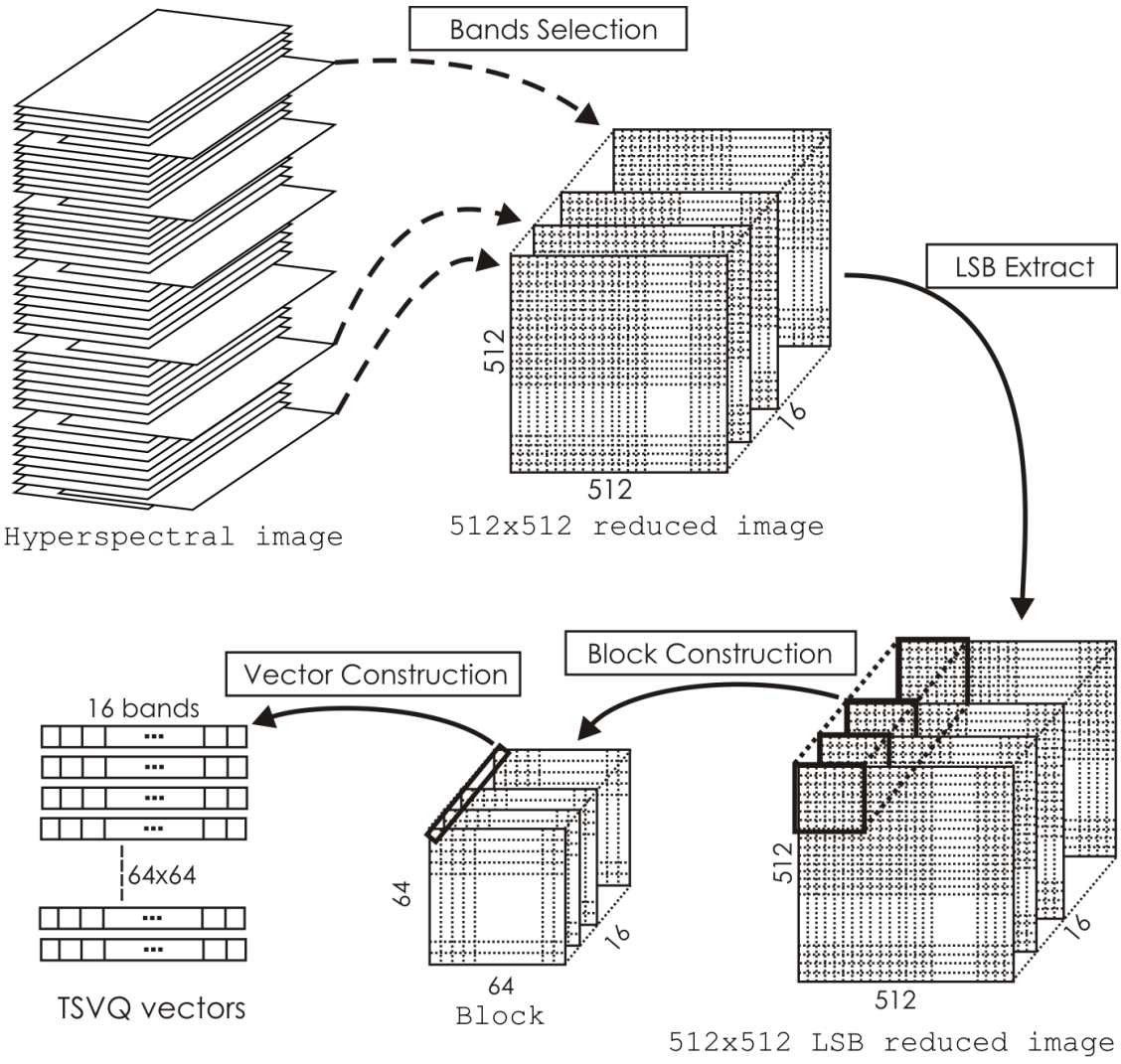

Figure 4: Band selection, block construction and generation of the signature vectors.

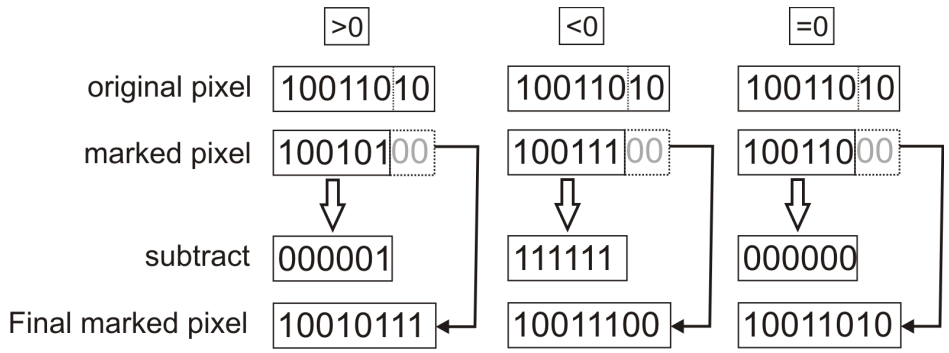

Figure 5: LSB restore scheme.

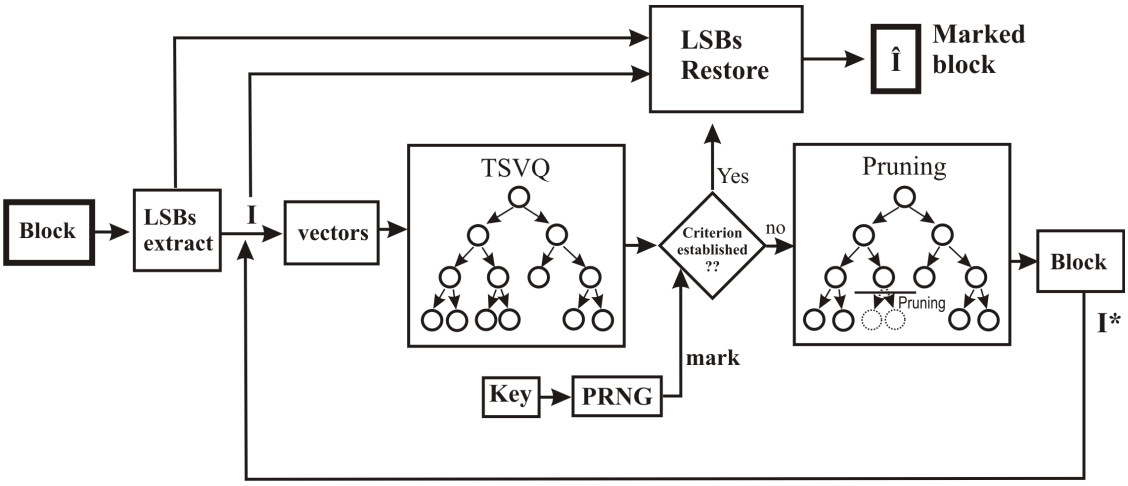

Figure 6: Block diagram of the embedding process.

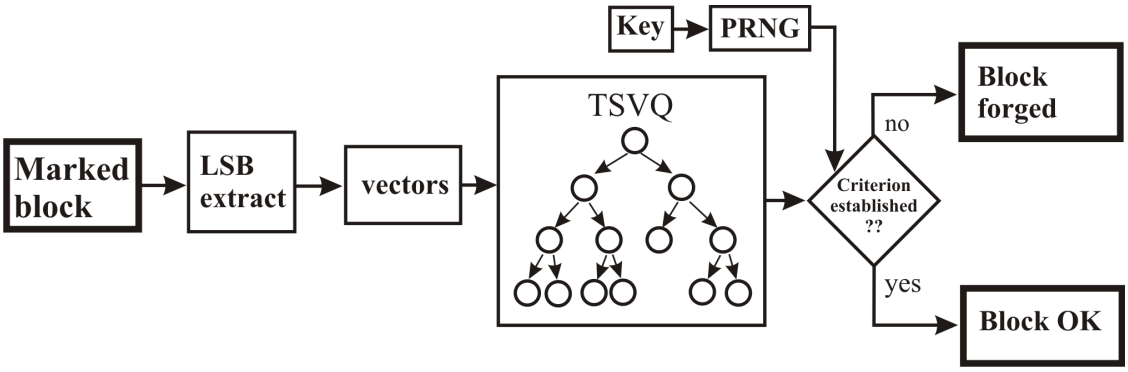

Figure 7: Block diagram of the detection process.

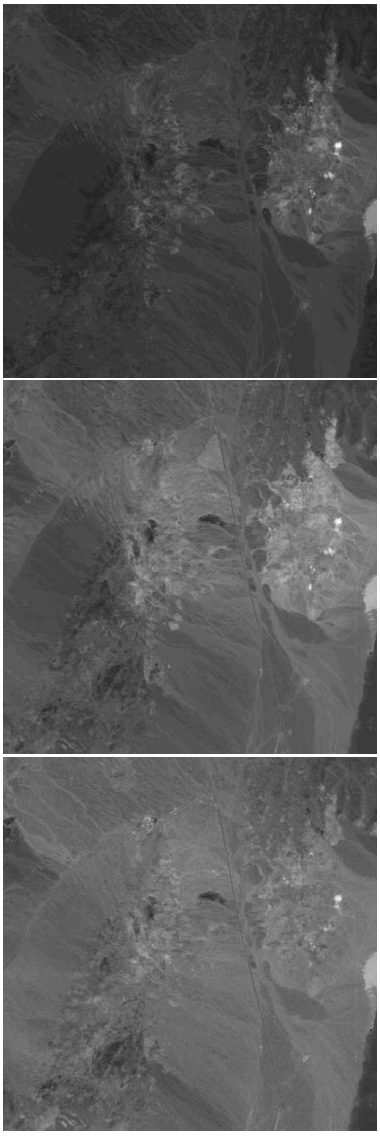

Figure 8: Example of the multispectral image bands 2, 7 and 12.

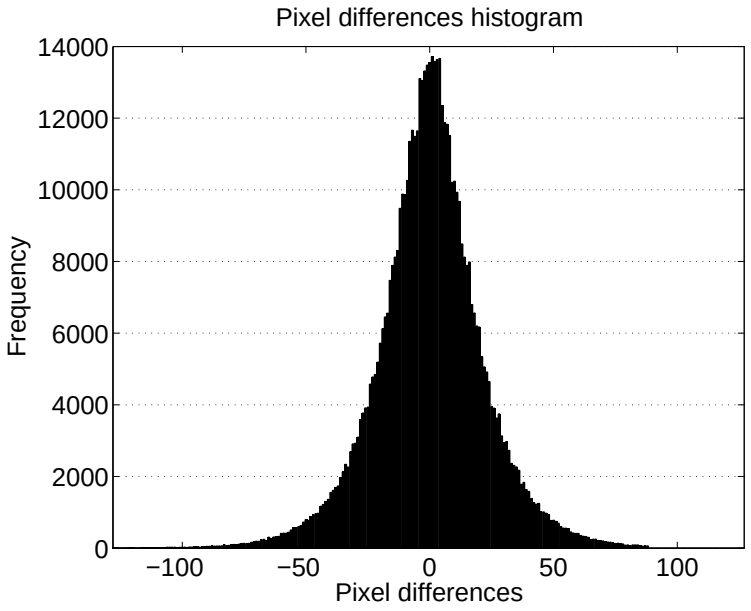

Figure 9: Histogram of the embedding distortion for the modified pixels.

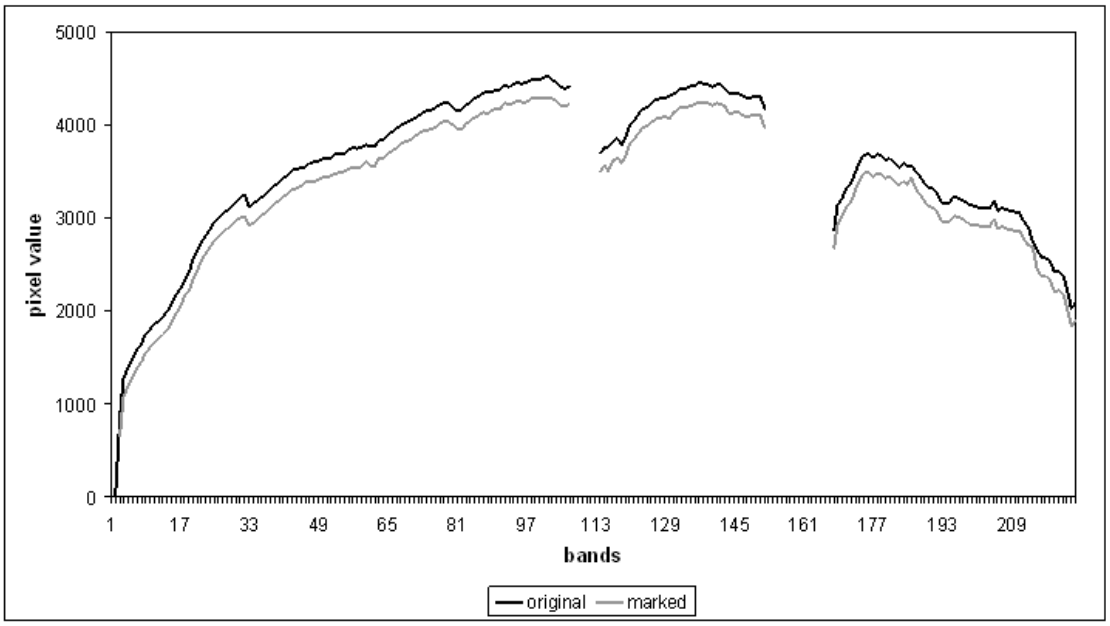

Figure 10: Sample difference of the original and marked (shifted) signatures for a modified pixel.

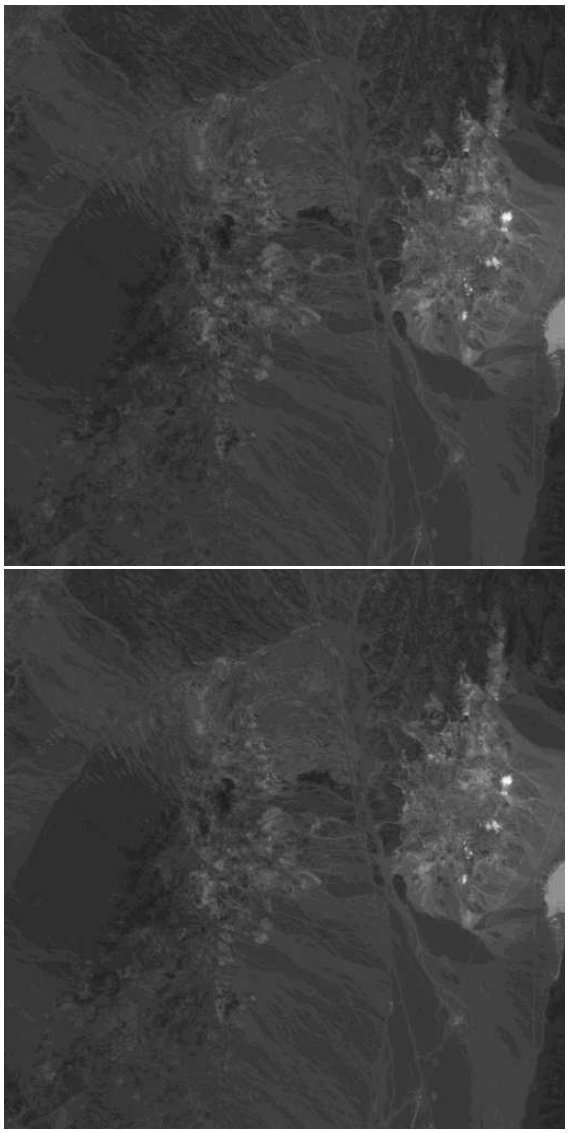

Figure 11: Original (top) and marked (bottom) images for band 2.

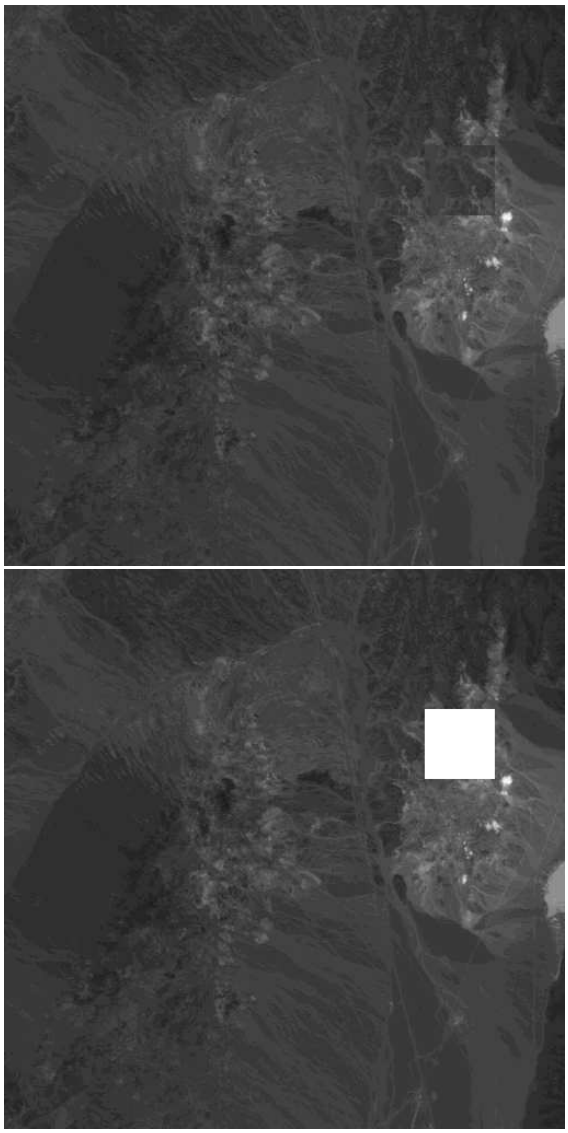

Figure 12: Tampered image (top) and tamper location (bottom).